III International Conference on Particle-based Methods – Fundamentals and Applications
PARTICLES 2013
M. Bischoff, E. Oñate, D.R.J. Owen, E. Ramm & P. Wriggers (Eds)

NOVEL MS (MUNJIZASCHIAVA) CONTACT DETECTION ALGORITHM ON MULTICORE PC

GUILLERMO G. SCHIAVA D'ALBANO[1], ANTE MUNJIZA[2] AND TOMAS LUKAS[3]

[1]School of Engineering and Material Science (SEMS) Queen Mary University of London
Mile End Road, E1 4NS, London, UK
E-mail: g.g.schiava@qmul.ac.uk Web page: www.sems.qmul.ac.uk

[2] School of Engineering and Material Science (SEMS) Queen Mary University of London
Mile End Road, E1 4NS, London, UK
E-mail: a.munjiza@qmul.ac.uk Web page: http://webspace.qmul.ac.uk/amunjiza/

[3] School of Engineering and Material Science (SEMS) Queen Mary University of London
Mile End Road, E1 4NS, London, UK
E-mail: t.lukas@qmul.ac.uk Web page: www.sems.qmul.ac.uk

Key words: Contact Detection, FDEM, Binary Tree, DEM-FEM, Multicore PC, MPI.

Abstract. A novel Contact Detection (CD) algorithm for Body-Point interaction for Combined Finite Discrete Element Method (FDEM) has been developed and tested on a Multicore PC. It is based on the decomposition of the domain into identical cubical cells and on the Binary Tree. The description of the algorithm and some applications are described in this paper.

1 INTRODUCTION

The Combined Finite Discrete Element Method is a relatively new computational tool developed in the mid '90s by Professor Munjiza. FDEM allows the simulations of complex phenomena in which the interaction of independent entities called Discrete Elements (DEs) collides, deform and fracture as a consequence of the virtual experimentation without requiring any predefined fracture paths from the user.

Each DE is made of *n=1,2,..,N* Finite Elements (FEs), "glued" together using joints. Fracture plastic behaviours are incorporated into the joints using different mathematical models such as the Smeared Crack Model[1].

Each FE deforms independently, taking into account forces from the joints, contact detection, etc. The FE governing equation is given by

$$\boldsymbol{M\ddot{u}} + \boldsymbol{C\dot{u}} + \boldsymbol{F_{int}} + \boldsymbol{F_{ext}} - \boldsymbol{F_{con}} - \boldsymbol{F_{jnt}} = \boldsymbol{0} \tag{1}$$

where $\boldsymbol{M}$ is the mass matrix, $\boldsymbol{C}$ is the damping matrix, $\boldsymbol{\ddot{u}}$ is the acceleration, $\boldsymbol{\dot{u}}$ is the velocity, $\boldsymbol{F_{int}}$ are the internal forces, $\boldsymbol{F_{ext}}$ are the external forces, $\boldsymbol{F_{con}}$ are the contact forces and $\boldsymbol{F_{jnt}}$ are the joint forces.

As a computational tool FDEM presents advantages when compared with more traditional methods such as the Finite Element Method (FEM) and to other Discrete Element Methods, such as Discontinuous Deformation Analysis (DDA). In FDEM the equations to solve are local to each FE, so there is no need to build a global system of equations. This simplifies the use of parallel tools such as the Message Passing Interface (MPI) using spatial domain decomposition[3].

The search and calculus of all the contact interactions between each one of the independent DEs represent a challenging task in CPU and algorithmic terms. Interactions line/tetrahedron, triangle/tetrahedron, tetrahedron/tetrahedron imply different interaction algorithms for each particular case. If on the contrary all interactions are simplified as point/tetrahedron only one algorithm can account for all kinds of interactions[4]. In this context a novel contact detection algorithm denominated MunjizaSchiava (MS) for Body-Point has been developed[2].

2 MS ALGORITHM

The MS algorithm is based on the Binary Tree[5] (BT), the Balance Binary Tree Schiava[2] (BBTS), the MunjizaRougier[6] (MR) algorithm and the discretisation of the space $\mathbb{R}$ into identical squares in 2D and boxes in 3D, denominated $\mathbb{R}_s$. In the MS algorithm contact detection is not performed every time step, rather it is only performed in the case that:

- objects are sent/received between processes event trigger by the parallel algorithm.
- the user defined number of steps to perform CD (n_{CD}) is equal to the number of steps since the last CD i.e. $n_{CD} = n_{\Delta CD}$

All the contact couples found during the CD are stored in a database that is called upon for each time step to perform an interaction, as shown in Figure 1.

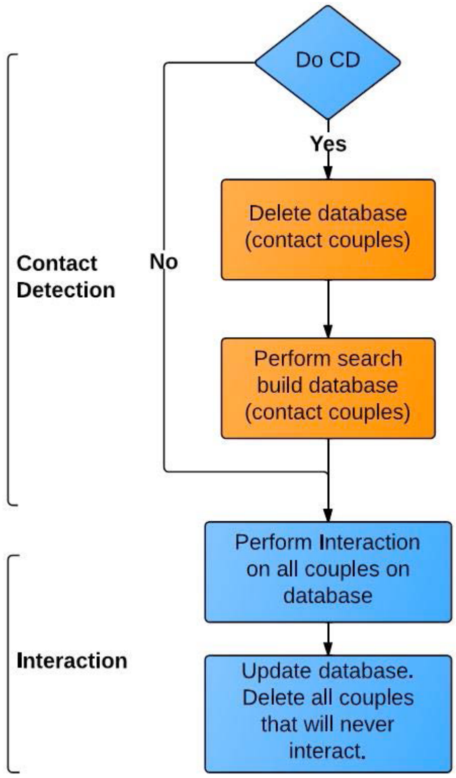


Figure 1: CD and Interaction flow diagram. Image reproduced from Schiava[2]

As contact detection is not performed each time step, the dimensions of all the contactors (finite elements) are increased, as shown in Figure 2, to account for possible displacements of targets (points) between calls to CD by

$$\Delta_{buf} = 2 n_{CD} v_{max} \Delta t \tag{2}$$

where n_{CD} is the number of steps between CD, v_{max} is the maximum velocity of any DE and Δt is the time step.

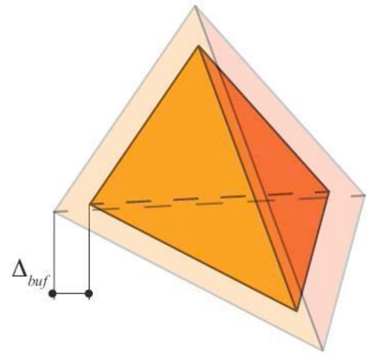


Figure 2: FE with its dimensions increased by Δ_{buf}. Image reproduced from Schiava[2]

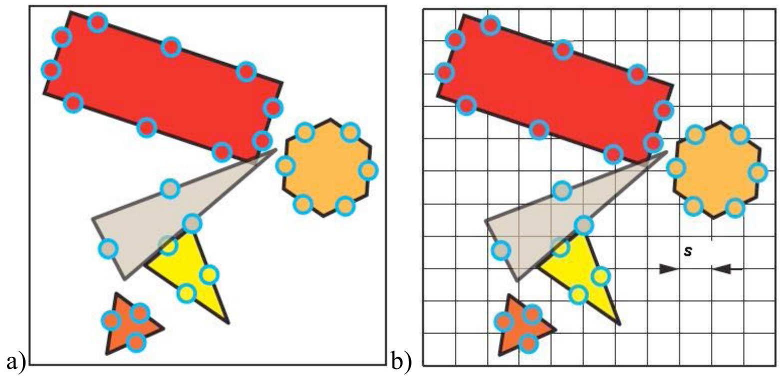


Figure 3:Domain decomposition. a) $\mathbb{R}$. b) $\mathbb{R}_s$. Images reproduced from Schiava[2]

The size s of the squares in 2D or the boxes in 3D of $\mathbb{R}_s$ is chosen by the user and is given by

$$s \propto \Delta_{buf} \tag{3}$$

Mapping from $\mathbb{R}$ to $\mathbb{R}_s$ is performed by

$$i_x = Int\left(\frac{x}{s}\right) \tag{4}$$
$$i_y = Int\left(\frac{y}{s}\right)$$
$$i_z = Int\left(\frac{z}{s}\right)$$

where x, y and z are the coordinates in $\mathbb{R}$, s is the cell size and i_x, i_y and i_z are the integerised coordinates in $\mathbb{R}_s$. The two different domains $\mathbb{R}$ and $\mathbb{R}_s$ are better described by Figure 3a and Figure 3b where different FEs are shown with their interaction points before and after the mapping.

Each FE is mapped/transformed in a rectangle in 2D or a rectangular cuboid in 3D, thus simplifying the process of contact detection. This mapping is not CPU intensive as only four integerised coordinates are required to be calculated in 2D and six integerised coordinate are required in 3D, as shown in equation 5 and equation 6.

$$i_{x_{min}} = Int\left(\frac{x_{min} - \Delta_{buf}}{s}\right)$$
$$i_{y_{min}} = Int\left(\frac{y_{min} - \Delta_{buf}}{s}\right)$$
$$i_{z_{min}} = Int\left(\frac{z_{min} - \Delta_{buf}}{s}\right) \tag{5}$$

$$i_{x_{max}} = Int\left(\frac{x_{max} - \Delta_{buf}}{s}\right)$$
$$i_{y_{max}} = Int\left(\frac{y_{max} - \Delta_{buf}}{s}\right)$$
$$i_{z_{max}} = Int\left(\frac{z_{max} - \Delta_{buf}}{s}\right) \tag{6}$$

The area in 2D and the volume in 3D delimitated by the minimum and maximum integerised coordinates is denominated $\mathbb{R}_{s\_CD}$. On this subdomain CD is going to be performed between a FE and all interaction points inside $\mathbb{R}_{s\_CD}$, as shown in Figure 4.

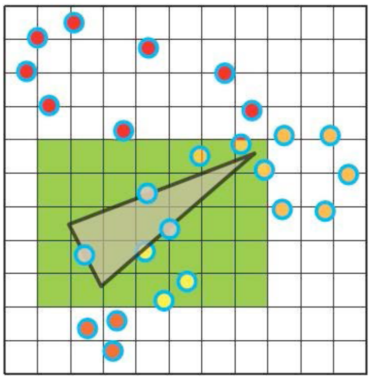


Figure 4:Subdomain $\mathbb{R}_{s\_CD}$ highlighted in green. Image reproduced from Schiava[2]

3 MS-TREE

The MS-tree is based on the BBTS binary tree with the addition of some improvements in order to avoid searching for empty cells during CD. The BBTS objects are loaded using the spatial ordering criterion[6] shown in equation 7 which states that a node i with key value V_i is bigger than a node j with key value V_j if $V_i > V_j$. The spatial ordering criterion can be seen as a hierarchy criterion of contact points as shown in Figure 5a.

To fully take advantage of this criterion two modifications are made to the BBTS:

- more than one objects with the same key value $V(i_x, i_y, i_z)$ is allowed to coexist
- a new pointer to the smallest greater object (*next* object) is added to each node, as shown in Figure 5b

$$\left\{\left[\left(i_{z_i} > i_{z_j}\right)\right] or \left[\left(i_{z_i} == i_{z_j}\right) and \left(i_{y_i} > i_{y_j}\right)\right]\right\}$$
$$or \left\{\left[\left(i_{z_i} == i_{z_j}\right) and \left(i_{y_i} == i_{y_j}\right) and \left(i_{x_i} > i_{x_j}\right)\right]\right\} \tag{7}$$

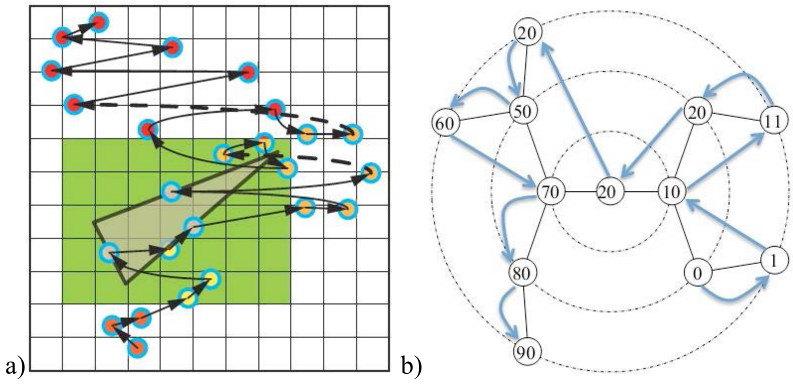


Figure 5 a) Hierarchy of contact points. b) Binary bush[7] of MS-tree: pointers to the *next* node are highlighted in light blue. Images reproduced from Schiava[2]

Algorithm 1: MS-load, add a new node to the tree using the spatial orderin criterium shown in equation 7. On the newly added node, set the pointer to the next object. Algorithm reproduced from Schiava[2]

$Nod_{smaGre} = \text{Null}$ ▷ Pointer smallest greater node
$Nod_{bigSma} = \text{Null}$ ▷ Pointer biggest smaller node
$Nod_{prv} = \text{Null}$ ▷ Pointer node previous
$Nod_{cur} = BT \rightarrow GetFrs()$ ▷ Get first node on tree
while $(Nod_{cur} != \text{Null})$ **do**
  $Nod_{prv} = Nod_{cur}$
  if $BT \rightarrow IsGreThan(Nod_{cur}, V_{new})$ **then** ▷ IsGreaterThan i.e. $(V_{cur} > V_{new})$
    $Nod_{smaGre} = Nod_{cur}$
    $Nod_{cur} = BT \rightarrow GetRigSon(Nod_{cur})$ ▷ go to the rigth son (R_s)
  else ▷ i.e. equal or less than V_{new}
    $Nod_{bigSma} = Nod_{cur}$
    $Nod_{cur} = BT \rightarrow GetLefSon(Nod_{cur})$ ▷ go to the left son (L_s)
  end if
end while
$Nod_{cur} = Nod_{prv}$
create Nod_{new} set variables on node
if $(Nod_{bigSma} != \text{Null})$ **then**
  $Nod_{bigSma} \rightarrow AddNxt(Nod_{new})$
else
  $Nod_{new} \rightarrow AddNxt(Nod_{smaGre})$
end if

4 CD-MS

The contact detection MS procedure is divided into four steps:

1. Clean database (contact couples)
2. MS-load: load targets (interaction points) into MS-tree
3. For each contactor (FE) perform MS-search and add new contact detection couples to database.
4. Clean MS-tree

4.1 MS-load

Each of the interaction points are loaded to the MS-tree using the integerised coordinates calculated using equation 4 and the spatial ordering criterion shown in equation 7, using algorithm 1.

4.1 MS-search

The key idea behind the MS-search algorithm is to take advantage of the spatial ordering criterion avoiding unnecessary operations on empty cells. This algorithm is explained using a simple example and the 2D algorithm 2. For the 3D algorithm please refer to Schiava[2].

Algorithm 2: MS-search. Algorithm reproduced from Schiava[2]

integer iLv_x_{min}, iLv_y_{min} ▷ Min integerised coordinate of contactor Con_{Cur} on $\mathbb{R}_s$
integer iLv_x_{max}, iLv_y_{max} ▷ Max integerised coordinate of contactor Con_{cur} on $\mathbb{R}_s$
integer iLv_x_{cur}, iLv_y_{cur} ▷ Current integerised coordinates
Con_{cur} ▷ Pointer to contactor (finite element) current
$Nod_{cur} =$ Null ▷ Pointer to current node
$Nod_{lst} =$ Null ▷ Pointer to last node
$Nod_{cur} = BT \to FndSmaGreY\,(iLv\_y_{min} - 1)$ ▷ FindSmallestGreater_Y
$iLv\_y_{cur} = BT \to GetY\,(Nod_{cur})$ ▷ GetYcoordiante of node current
while $(iLv\_y_{cur} < iLv\_y_{max})$ **do**
    $Nod_{cur} = BT \to FndSmaGreYX\,(iLv\_y_{cur}, (iLv\_x_{min} - 1))$
    $Nod_{lst} = BT \to FndSmaGreYX\,(iLv\_y_{cur}, iLv\_x_{max})$
    while $(Nod_{cur}! = Nod_{lst})$ **do**
        $vS\_Interaction\,(Con_{cur}, Nod_{cur})$ ▷ If there is contact add to data base
        $Nod_{cur} = BT \to GetNxt\,(Nod_{cur})$ ▷ GetNextNodeOfCurrentNode
    end while
    $Nod_{cur} = BT \to FndSmaGreY\,(iLv\_y_{cur})$ ▷ FindSmallestGreater_Y
    $iLv\_y_{cur} = BT \to GetY\,(Nod_{cur})$ ▷ GetYcoordiante of node current
end while

After all interactions points shown in Figure 6a have been loaded onto the MS-tree, it is possible to consider the MS-tree as both a binary tree and as an ordered single connected list as shown in Figure 6b.

The first step is to find the first "non empty row y" between the limits of $\mathbb{R}_{s\_CD}$ i.e. find the first non empty row greater than the row 0, highlighted by a red square in Figure 6a. Subsequently, one must carry out a search on the same row for the first non-empty node inside the domain $\mathbb{R}_{s\_CD}$ and the first smallest greater node outside $\mathbb{R}_{s\_CD}$ on the same row, as shown in Figure 7a and Figure 7b. These two nodes are the first node to perform CD (node 2 in Figure 7b) and the first node to skip (node 1 Figure 7b). It is worth noting that no searches have been produced on the empty cells at any point.

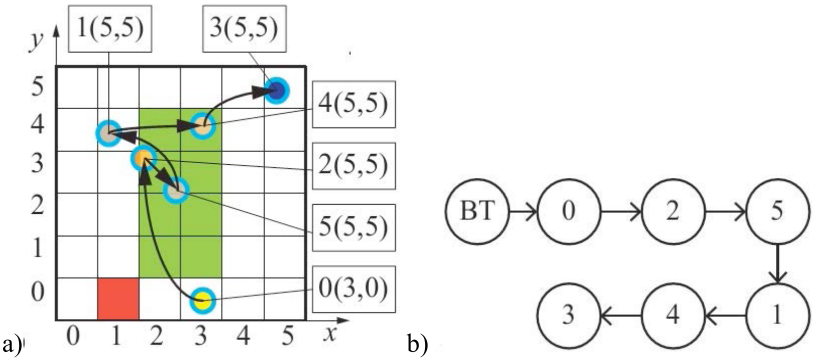


Figure 6 $\mathbb{R}_s$ domain. a) The $\mathbb{R}_{s\_CD}$ domain of a generic FE is highlighted in green. Each interaction point is pointing to the next smaller greater point. b) A single connected list produced by the MS-tree. Images reproduced from Schiava[2]

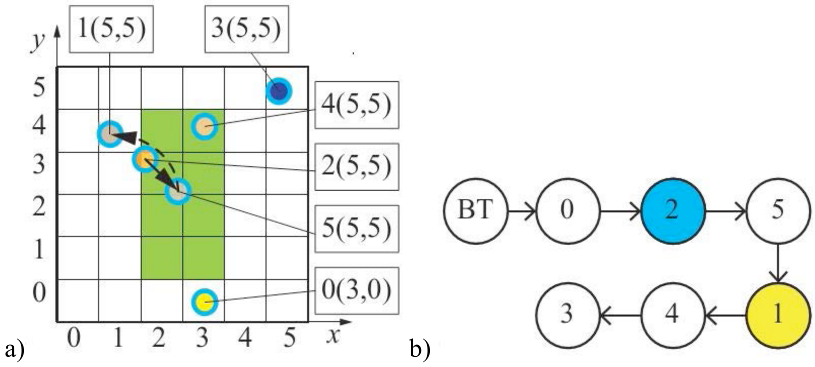


Figure 7 Hierarchy of nodes to perform CD a) On $\mathbb{R}_{s\_CD}$. b) On a single connected list produced by the MS-tree. The nodes to check for CD are 2 and 5. Images reproduced from Schiava[2]

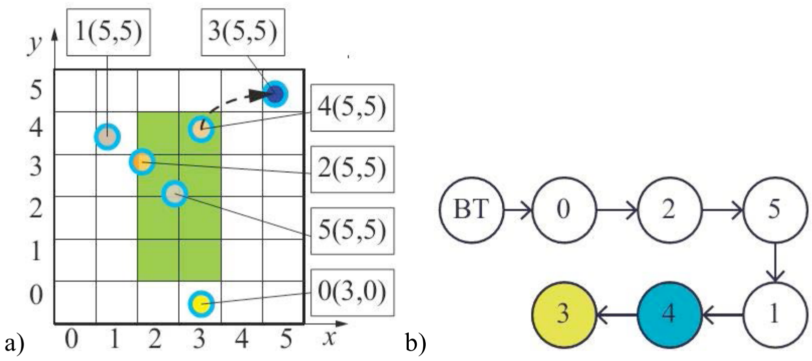
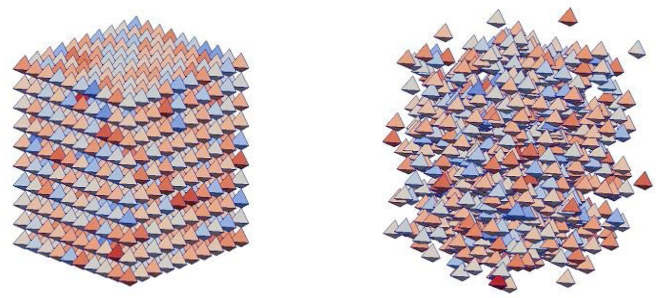


Figure 8 Hierarchy of nodes to perform CD a) On $\mathbb{R}_{s\_CD}$. b) On a single connected list produced by the MS-tree. The node to check for CD is 4. Images reproduced from Schiava[2]

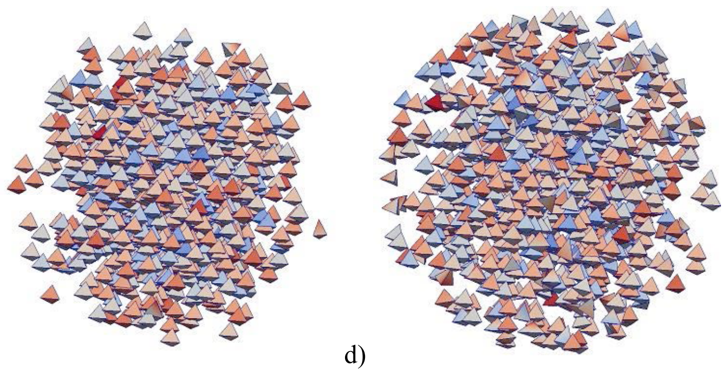
The procedure is repeated while nodes are detected inside $\mathbb{R}_{s\_CD}$ as shown in Figure 8a and Figure 8b.

5 TEST

This novel algorithm is tested[2] on a Multicore PC DELL Precision T5400 with one processor of four cores with 32 G of RAM, in sequential (one core) and in parallel (two and four cores). The test consists of a raster of DEs made of one FE. Random velocities from 0 m/s to 137 m/s are assigned to each FE. The material properties are: elastic modulus E=4 MPa, Poisson's ratio ν=0.45 and penalty σ_p=400 MPa. A spherical boundary with penalty σ_p=400 MPa is imposed to keep all FEs interacting with each other. Each FE has one interaction point centred on each of its faces.



a) b)



c) d)

Figure 9. Raster 1000 DEs, four processors. a) Time 0.0 ms. b) Time 0.32 ms. c)Time 0.60 ms. d)Time 1.01ms Images reproduced from Schiava[2]

The development of the simulation are shown in Figure 9a to Figure 9d while the CD-MS is tested up to 1.331 million DEs and 5.324 million interaction points, the total time for the sequential and parallel virtual experimentation are shown in Figure 10.

CD-MS, as is based on the BT for performing searches is not a linear algorithm. Still in the tested range present a quasi-linear behaviour.

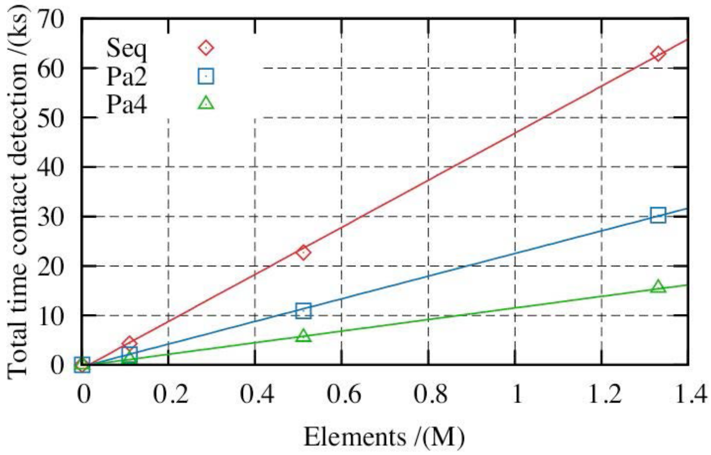


Figure 10 Total CD-MS time. Image reproduced from Schiava[2]

5 COLLAPSE OF A COOLING TOWER

Dynamic simulations are among the most CPU demanding FDEM virtual experimentations. This example[2] consists in a hyperboloid-cooling tower made of concrete as shown in Figure 11a. The material is linear elastic with elastic modulus E= 35 GPa, strain energy release rate G_f=147.5 N/m, tensile strength f_t=6.3 MPa density ρ=2400 kg/m^3 and Poisson's ratio ν=0.15. All the interactions are calculated as non-elastic[2]. The domain is discretised in four processes, as shown in Figure 11b.

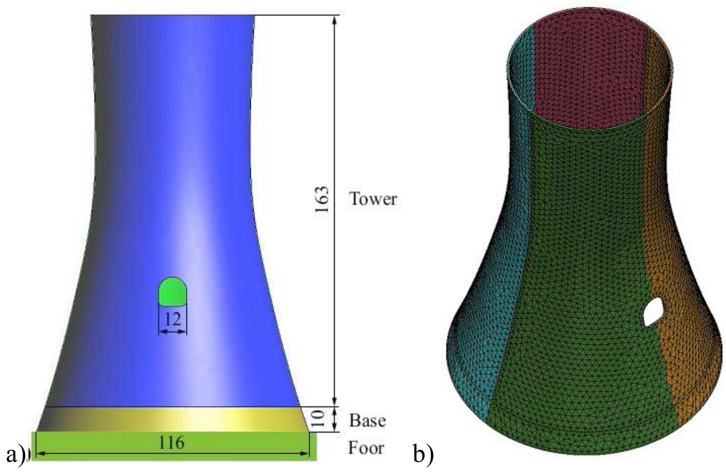


Figure 11 Cooling tower a) Dimensions (m). b) Processes topology and mesh. Images reproduced from Schiava[2]

The virtual experiment is split into two steps, the first step with boundary conditions shown in Figure 12a consists in the loading part in which the gravity is gradually increased and the tower is allowed to deform until the kinetic energy is nearly equal to zero. The second part consists in the collapse of the tower, as there are no more fixed boundary conditions and the front base is fractured at a time equal to zero, as shown in Figure 12b.

The tower collapse sequence is shown in Figure 13a to Figure 13d.

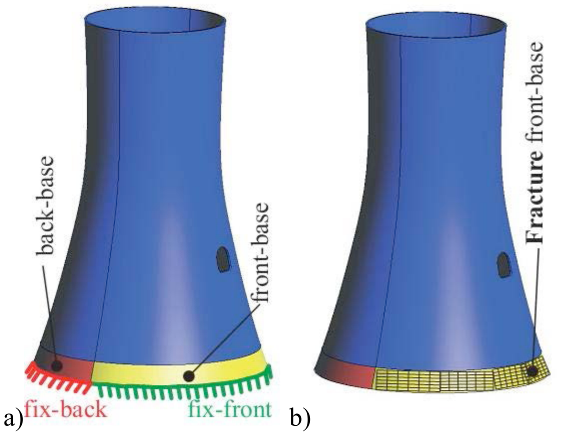


Figure 12 Coolling tower a) Dimensions (m). b) Processes topology and mesh. Images reproduced from Schiava[2]

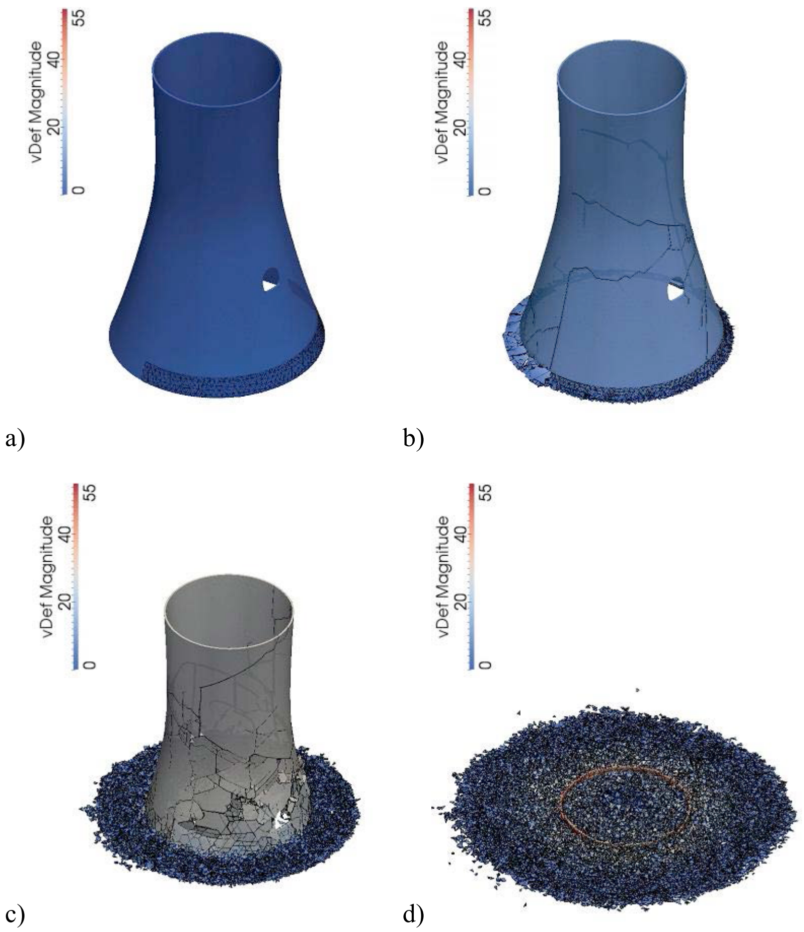


Figure 13 Tower collapse a) t= 0.00 s. b) t= 0.9952 s. c) t= 2.9952 s. c) t= 5.9936 s. Images reproduced from Schiava[2]

3 CONCLUSIONS

Novel contact detection for body-point was presented and tested on a Multicore PC. The good performance of the algorithm was tested with more than a million discrete elements.

Further improvements are possible if the domain $\mathbb{R}_{s\_CD}$ implemented in the current work as shown in Figure 14a is reduced as shown in Figure 14b. In this way unnecessary operations during contact detection can be avoided.

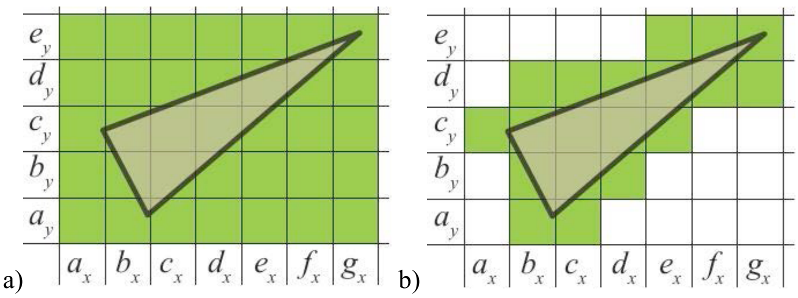


Figure 14 Domain $\mathbb{R}_{s\_CD}$ a) Implemented in the current MS algorithm. b) Propose improvement. Images reproduced from Schiava[2]

REFERENCES

[1] Munjiza, A. *The Combined Finite Discrete Element Method*. Wiley, (2004).

[2] Schiava D'Albano G.G. *Computational and Algorithmic Solutions for Large Scale Combined Finite-Discrete Elements Simulations*. PhD Thesis, (2013).

[3] Lukas, T. *Space Decomposition Based Parallelization Solutions for the Combined Finite-Discrete Element Method in 2D*. PhD Thesis, (2013).

[4] Munjiza, A. and Knight, E. and Rougier E. *Computational Mechanics o Dicontinua*. Wiley, (2011).

[5] G. Knott A. Balanced tree storage and retrieval algorithm. *Annual ACM conference on Research and Development in Information Retrieval*, 175-196 (1971).

[6] Munjiza, A. and Rougier, E. and John, N. MR linear contact detection algorithm. *Int. J. Num. Meth. Engng* (2006) **66**:46-71.

[7] Munjiza, A. and John, N. Towards one billion Particle System. *3rd MPI conference on computational fluid and solid mechanics* (2005).